

IN THE CLAIMS:

Please amend claims 1, 5-6, 11-12, 14 and 26-28 as follows:

1. (Currently Amended) A load distribution method adapted-by in a client-server system including a plurality of clients, ~~[[and]]~~ a server cluster and a storage device which is coupled to said serves cluster and stores data to be used by at least one of the clients, said server cluster including a plurality of servers each of which includes a cache memory, and said server cluster being used for processing requests related to the data in the storage device from ~~made by~~ said clients and ~~allows~~ allowing a number of said servers to be changed dynamically, comprising the steps of:
 - detecting, by one of clients, a change of the number of servers forming said server cluster;
 - setting, by said one client, an allocation of requests transmissible out to a newly added server at an initial amount smaller than amounts set for the remaining servers in the server cluster, right after detecting an increase in the number of servers;
 - monitoring performance of the newly added server;
 - determining whether a cache hit rate for a cache memory in the newly added server is beyond a predetermined value based upon the monitored performance;
 - increasing an amount of requests to be allocated to the newly added server if the cache hit rate is beyond the predetermined value;
 - transmitting out requests to said servers on the basis of said set allocation, if said increase in the number of servers is detected; and
 - receiving, by said one client, responses to the requests from said servers.
2. (Previously Presented) A load distribution method according to claim 1, further comprising a step of increasing, by said one client, said amount of allocation of requests transmissible out to said newly added server with the lapse of time since said newly added server is detected.
3. (Previously Presented) A load distribution method according to claim 1, wherein said detection of an increase in said number of said servers is used as a trigger of each of said clients to set said allocation of requests transmissible out to said newly added server at said amount smaller than said amounts set for said remaining servers.

4. (Previously Presented) A load distribution method according to claim 1, further comprising the steps of:
 - acquiring, by said one client, information on a performance of said newly added server, if said newly added server is detected; and
 - setting, by said one client, said allocation of requests transmissible out to said newly added server on the basis of said acquired information.
5. (Currently Amended) A load distribution method according to claim 1, ~~further comprising the steps of:~~ wherein the step of monitoring performs acquiring performance information of said newly added server; ~~and setting, by said one client, said allocation of requests transmissible out to said newly added server on the basis of said acquired information.~~
6. (Currently Amended) A load distribution method according to claim 5, wherein said performance information of said newly added server includes at least one of ~~[[a]]~~ the cache hit rate, a cache utilization ratio and a number of requests each waiting for a processing turn, and
 - said amounts of allocation of requests set for the servers including said newly added server are set based upon said at least one of ~~[[a]]~~ the cache hit rate, a cache utilization ratio and a number of requests each waiting for a processing turn.
7. (Previously Presented) A load distribution method according to claim 1 wherein:
 - said client-server system has a management server for managing the number of servers composing said server cluster; and
 - a notice received from said management server as a notice of an increase in said number of said servers is used as a trigger of each of said clients to set said allocation of requests transmissible out to said newly added server at said amount smaller than said amounts set for said remaining servers.
8. (Previously Presented) A load distribution method according to claim 1, wherein said client-server system has a management server, and the method further comprising the steps of:

acquiring, by said one client, information on a performance of each of said servers from the management server; and

setting, by said one client, said allocation of requests transmissible out to said newly added server on the basis of said acquired information.

9. (Previously Presented) A load distribution method according to claim 1, further comprising a step of:

setting, by said one client, said allocation of requests transmissible out to said newly added server by setting a number of connections for communications with said servers.

10. (Previously Presented) A load distribution method according to claim 1, further comprising a step of:

setting, by said one client, an allocation of requests transmissible out to each of said servers by changing quotas each set for every individual one of said servers as an allotment of requests transmissible out to said individual server.

11. (Currently Amended) A load distribution method according to claim 10, ~~wherein said client-server system has storage apparatus connected to said servers, and the method further comprising the steps of:~~

holding, by said servers, directory information indicating storage locations of files stored in said storage apparatus; and

setting, by said one client, said allocation of requests transmissible out to each of said servers by changing quotas each provided for every individual one of said servers as an allotment of said directory information stored in said individual server where said allotment of said directory information storable in said individual server represents an allotment of requests transmissible out to said individual server.

12. (Currently Amended) A client-server system, comprising:

a plurality of clients;

a storage device for storing data used by said clients; and

a server cluster which is formed by a plurality of servers each of which includes a cache memory to hold said data temporarily, each of said servers being

coupled to said storage device and used for processing requests made by said clients,
wherein each of said clients includes:

a server-count detection unit which detects a number of the servers included in said server cluster;

an allocation setting unit which sets an allocation of requests transmissible out to a newly added server as an initial amount smaller than amounts set for the remaining servers, if said server-count detection unit detects an increase in the number of servers;

a monitoring unit which monitors performance of the newly added server, determines whether a cache hit rate for a cache memory in the newly added server is beyond predetermined value based upon the monitored performance, and increases an amount of requests to be allocated to the newly added server if the cache hit rate is beyond the predetermined value;

a request distribution unit which transmits out requests to each of said servers on the basis of the allocations set by said allocation setting unit, and

a receiving unit which receives responses to said requests from said servers.

13. (Previously Presented) A client-server system according to claim 12, wherein:

each of said clients has an allotment-holding unit for holding an allotment set for every individual one of said servers as an allotment of requests transmissible out to said individual server; and

said allocation setting unit sets an allocation of requests transmissible out to each of said servers by changing quotas each set for every individual one of said servers as said allotment of requests transmissible out to said individual server.

14. (Currently Amended) A client-server system according to claim 13, ~~further comprising storage apparatus connected to said servers,~~

wherein each of said servers is provided with a directory-information-holding unit for holding directory information indicating storage locations of files stored in said storage apparatus;

said clients are provided with a management server for holding quotas each provided for every individual one of said servers as an allotment of said directory information storable in said individual server; and

said allocation setting unit sets said allocation of requests transmissible out to each of said servers by changing said quotas each provided for every individual one of said servers as an allotment of said directory information stored in said individual server.

15. (Cancelled)

16. (Previously Presented) A load distribution method according to claim 2, wherein the step of setting said allocation of requests transmissible out to said newly added server involves setting a number of connections for communications with said servers.

17-23. (Cancelled)

24. (Previously Presented) A client-server system according to claim 12, further comprising a management server for managing the number of servers;

wherein the clients receives a notice from said management server as a notice of an increase in said number of said servers, said notice is used as a trigger by each of said clients to set said allocation of requests transmissible out to said newly added server at said amount smaller than the amounts set for the remaining servers.

25. (Previously Presented) A client-server system according to claim 12, wherein each of the clients further includes an acquiring unit which acquires performance information of the servers, the information related to performance of request processing, and the performance information is used by the allocating setting unit.

26. (Currently Amended) A client-server system according to claim 25, wherein

~~the servers includes a cache which temporarily stores data,~~

the performance information includes performance of the cache including at least one of [[a]] the cache hit rate, a cache utilization ratio, and a number of requests each waiting for a processing turn, and

said amount of allocation of requests for the servers including said newly added server is set, by the allocating setting unit, on the basis of said at least one of [[a]] the cache hit rate, a cache utilization ratio, and a number of requests each waiting for a processing turn.

27. (Currently Amended) A client computer to be coupled to a server cluster including a plurality of servers which form into a server cluster each of which is coupled to a storage device and includes a cache memory, the storage device storing data used by the client computer, comprising:

a detecting unit which detects an increase in a number of servers of the server cluster;

a setting unit which sets an initial amount of requests to be sent to a new added sever from the client computer smaller than amounts of requests to be sent to the remaining severs from the client computer, if the new added sever is detected;

a monitoring unit which monitors performance of the newly added server, determines whether a cache hit rate for a cache memory in the newly added server is beyond a predetermined value based upon the monitored performance, and increases an amount of requests to be allocated to the newly added server if the cache hit rate is beyond the predetermined value;

a transmitting unit which transmits the requests set by the setting unit to the servers including the new added server; and

a receiving unit which receives responses to the requests from the servers.

28. (Currently Amended) A client computer according to claim 27, wherein the setting unit further changes said amount of the requests to be sent to the new added server based upon a time period since the new added server is detected till a current time point, after the monitoring unit increases the amount of requests to be allocated to the newly added server.

29. (Previously Presented) A client computer according to claim 27, further comprising an acquiring unit which acquires performance information of the servers,

wherein the servers includes a cache memory for temporarily storing data when executing the requests,

the acquiring unit acquires performance information of the cache memory including at least one of a cache bit rate, a cache utilization ratio, and a number of requests each waiting for an executing turn, and

the setting unit changes said amount of the requests to be sent to the new added server based upon the acquired performance information of the cache memory.